

Reinforcement Learning (AI644)

Prepared by Dr. Ashraf Elsayed

Lecture 2



Reinforcement Learning

Lecture 2

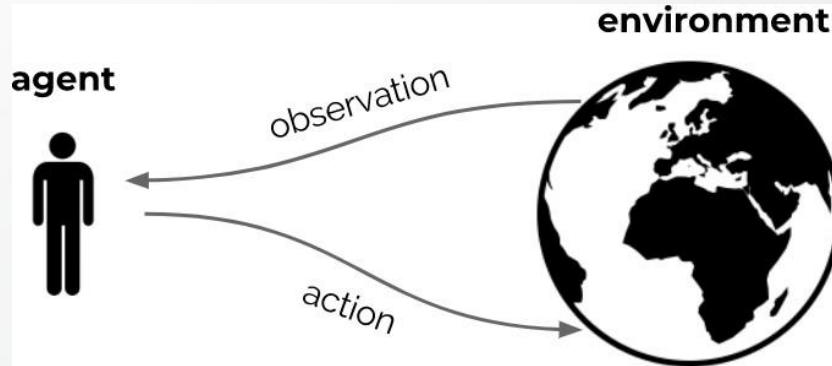
Prepared by
Dr. Ashraf Elsayed

Exploration and Exploitation

The objectives of this lecture is to:

- Understand Exploration and Exploitation
- Define and explain the trade off between Exploration and Exploitation.
- Understand The Multi-Armed Bandit.

Recap



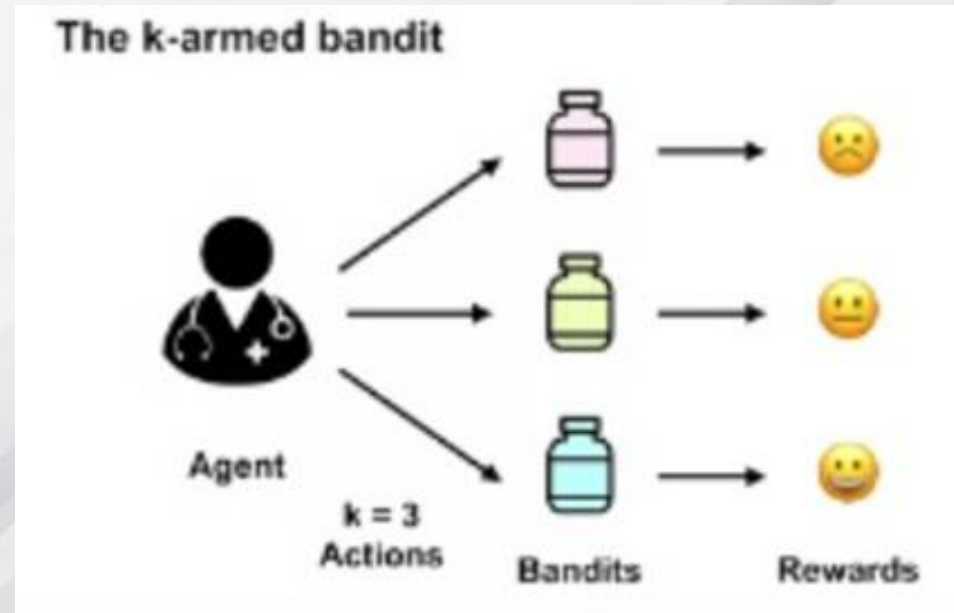
- Reinforcement learning is the science of learning to make decisions
- Agents can learn a **policy**, **value function** and/or a **model**
- The general problem involves taking into account **time** and **consequences**
- Decisions affect the **reward**, the **agent state**, and **environment state**
- The most important feature distinguishing reinforcement learning from other types of learning is that it uses training information that **evaluates** the actions taken rather than **instructs** by giving correct actions

Exploration vs. Exploitation

- In this lecture we study the **evaluative** aspect of reinforcement learning in a simplified setting, one that does not involve learning to act in more than **one situation**
- Learning agents need to trade off two things
 - **Exploitation**: Maximize performance based on current knowledge
 - **Exploration**: Increase knowledge
- We need to gather information to make the best overall decisions
- The best long-term strategy may involve short-term sacrifices

The Multi-Armed Bandit (Clinical Trial)

- **Sequential Decision Making with Evaluative Feedback**
 - Assume the problem where the doctor has to give medicine to a patient and based on which one he chooses, he will receive a reward (cure the patient).
 - For the doctor to decide which action is best, we must define the value of taking each action.
 - We call these values the *action values* or the *action value function*.



Action-Values

- The **action value** for action a is the expected reward

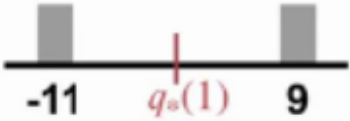
$$\begin{aligned} q_*(a) &\doteq \mathbb{E}[R_t | A_t = a] \forall a \in \{1, \dots, k\} \\ &= \sum_r p(r|a) r \end{aligned}$$

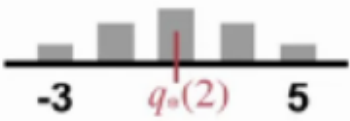
- The goal is to maximize the expected reward (**optimal value**)

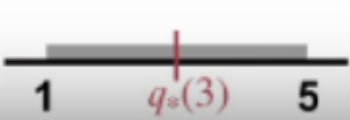
$$\operatorname{argmax}_a q_*(a)$$

Calculating Action-Values

Calculating $q_*(a)$


 $\rightarrow$

 $\rightarrow q_*(a) = .5 \times -11 + .5 \times 9$


 $\rightarrow$

 $\rightarrow q_*(a) = 1$


 $\rightarrow$

 $\rightarrow q_*(a) = 3$

Learning Action Values

- The value of an action is the expected reward when the action is taken

$$q_*(a) \doteq \mathbb{E}[R_t | A_t = a]$$

- $q_*(a)$ is not known, so we **estimate** it.
- The **estimated value for action** A is the sum of rewards observed when taking action A divided by the total number of times action A has been taken.

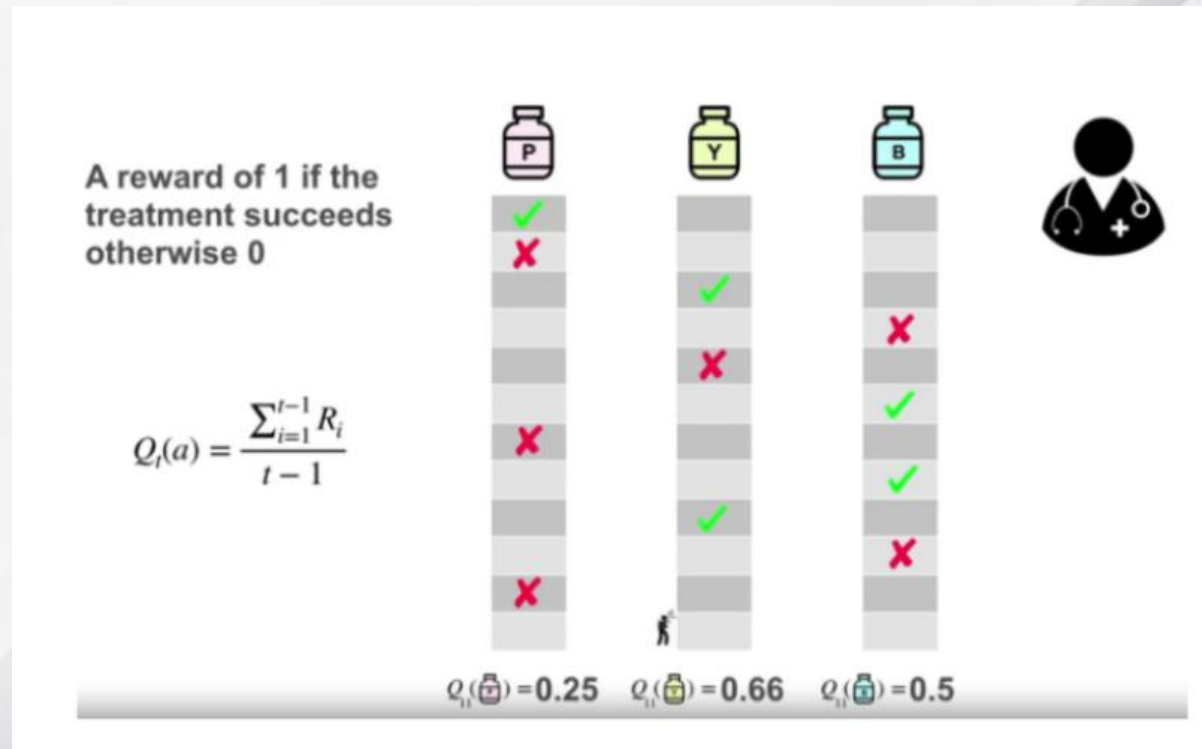
Sample-Average Method

- The estimate of $q_*(a)$ is denoted as $Q_t(a)$ and can be written as

$$Q_t(a) \doteq \frac{\text{sum of rewards when } a \text{ taken prior to } t}{\text{number of times } a \text{ taken prior to } t} = \frac{\sum_{i=1}^{t-1} R_i}{t-1}$$

Example of Sample-Average Method (Clinical Trial)

- Assume different attempts are made during the trial.
- As the experiment advances, the estimates of the Action Values became more accurate.

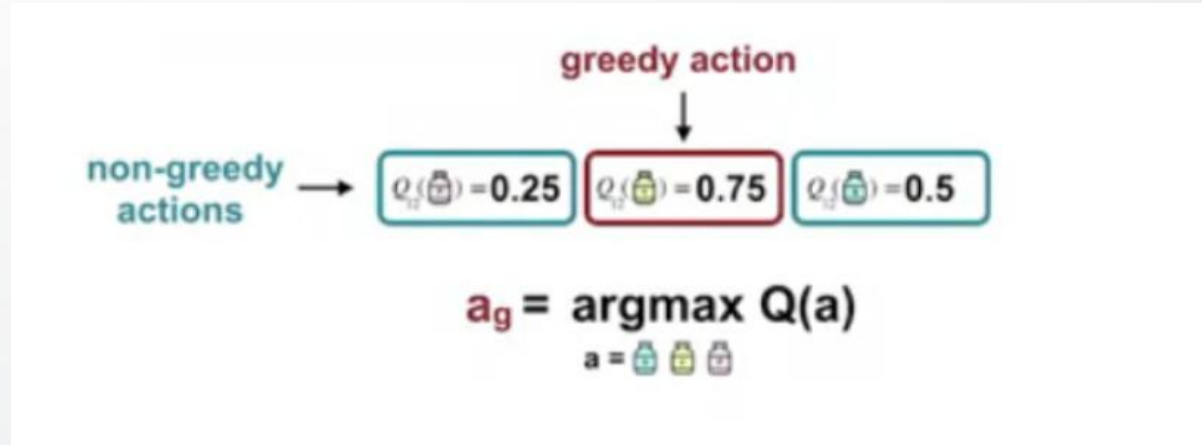


The greedy action

- In reality, our doctor would not randomly assign treatments to their patients.
- They would probably assign the treatment that they currently think is the best.
- We call this method of choosing actions *greedy*. The greedy action is the action that currently has the largest estimated value.
- Selecting the greedy action means the agent is *exploiting* its current knowledge.
- It is trying to get the most reward it can right now.
- Instead of behaving greedily most of the time, but every once in a while, say with small probability ϵ , instead select randomly from among all the actions with equal probability, independently of the action-value estimates.
- We call this method using this *near-greedy* action selection rule ϵ -*greedy* methods.

The exploration-exploitation dilemma

- We can compute the greedy action by taking the argmax_a of our estimated values.



- The agent cannot choose to both *explore* and *exploit* at the same time.
- This is one of the fundamental problems in reinforced learning.
- The *exploration-exploitation dilemma*.

Estimating Action Values Incrementally

- The action-value method we have discussed so far estimates action values as sample averages of observed rewards.
- These averages can be computed in a computationally efficient manner.
- The incremental update rule is derived as follows:

$$Q_n = \frac{1}{n-1} \sum_{i=1}^{n-1} R_i$$

The obvious implementation would be to maintain a record of all the rewards and then perform this computation whenever the estimated value was needed

$$Q_{n+1} = \frac{1}{n} \sum_{i=1}^n R_i = \frac{1}{n} \left(R_n + \sum_{i=1}^{n-1} R_i \right) = \frac{1}{n} \left(R_n + (n-1) \frac{1}{n-1} \sum_{i=1}^{n-1} R_i \right)$$

$$Q_{n+1} = \frac{1}{n} (R_n + (n-1) Q_n) = \frac{1}{n} (R_n + nQ_n - Q_n) = Q_n + \frac{1}{n} [R_n - Q_n]$$

Target

Incremental update rule

- The *error* in the estimate is the difference between the old estimate Q_n and the new target R_n (The new reward is our *target*).
- Taking a *step* towards that new target will create a *new estimate* that reduces our error.
- The size of the step is determined by our *step size* parameter and the *error* of our old estimate.

$$NewEstimate \leftarrow OldEstimate + Stepsize [Target - OldEstimate]$$

$$Q_{n+1} = Q_n + \alpha_n [R_n - Q_n]$$

- The step size α can be a function of n that produces a number from zero to one.

$$\alpha_n \rightarrow [0, 1]$$

- In the special case of the sample average, the step size is equal to one over n

$$\alpha_n = \frac{1}{n}$$

Bandit Algorithm Pseudocode

- In this Pseudocode for a complete bandit algorithm using incrementally computed sample averages and ϵ -greedy action selection is shown in the box below.
- The function **bandit(a)** is assumed to take an action and return a corresponding reward.
- α will be replaced by $\frac{1}{N(a)}$

A simple bandit algorithm

Initialize, for $a = 1$ to k :

$$Q(a) \leftarrow 0$$

$$N(a) \leftarrow 0$$

Loop forever:

$$A \leftarrow \begin{cases} \arg \max_a Q(a) & \text{with probability } 1 - \epsilon \\ \text{a random action} & \text{with probability } \epsilon \end{cases} \quad (\text{breaking ties randomly})$$

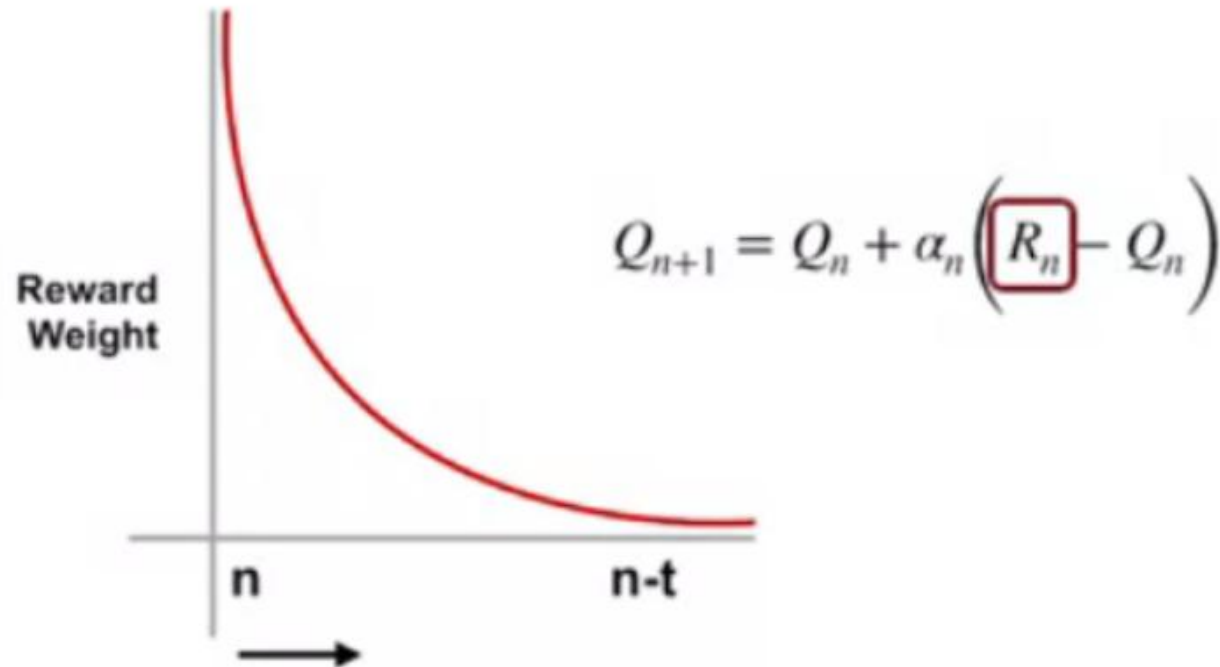
$$R \leftarrow \text{bandit}(A)$$

$$N(A) \leftarrow N(A) + 1$$

$$Q(A) \leftarrow Q(A) + \frac{1}{N(A)} [R - Q(A)]$$

Non-Stationary Bandit Problem

- These problems are like the bandit problems except the distribution of rewards changes with time.
- The doctor is unaware of this change but would like to adapt to it.
- One option is to use a fixed step size.
- If α_n is constant like 0.1, then the most recent rewards affect the estimate more than older rewards



Decaying past rewards

- The prove of the effectiveness of decaying the past rewards is shown as follows.

$$\begin{aligned}Q_{n+1} &= Q_n + \alpha [R_n - Q_n] \\&= \alpha R_n + (1 - \alpha) Q_n \\&= \alpha R_n + (1 - \alpha) [\alpha R_{n-1} + (1 - \alpha) Q_{n-1}] \\&= \alpha R_n + (1 - \alpha) \alpha R_{n-1} + (1 - \alpha)^2 Q_{n-1} \\&= \alpha R_n + (1 - \alpha) \alpha R_{n-1} + (1 - \alpha)^2 \alpha R_{n-2} + \\&\quad \dots + (1 - \alpha)^{n-1} \alpha R_1 + (1 - \alpha)^n Q_1 \\&= (1 - \alpha)^n \textcircled{Q_1} + \sum_{i=1}^n \alpha (1 - \alpha)^{n-i} R_i.\end{aligned}$$

The initial Action-Value



- The first term tells us that the contribution of Q_1 decreases exponentially with time.
- The second term tells us the *rewards* further back in time contribute exponentially less to the sum.
- Taken all together, we see that the influence of our initialization of Q goes to zero with more and more data.
- The *most recent rewards* contribute most to our current estimate.

What is the trade-off? (Example: Picking A Meal)

- $q(a)$ is the estimated value for picking a meal.
- $N(a)$ is the number of times you have picked that meal,
- $q_*(a)$ is the true value of each meal
- **Exploration** improves the knowledge for the long-term benefit.

Exploration and Exploitation

- **Exploration** - improve knowledge for long-term benefit



$$\begin{aligned} q(a) &= 2 \\ N(a) &= 2 \\ q_*(a) &= 3 \end{aligned}$$



$$\begin{aligned} q(a) &= 4 \\ N(a) &= 2 \\ q_*(a) &= 4 \end{aligned}$$



$$\begin{aligned} q(a) &= -1 \\ N(a) &= 1 \\ q_*(a) &= 2 \end{aligned}$$

What is the trade-off? (Example: Picking A Meal)

- **Exploitation** exploits the agent's current estimated values. It chooses the greedy action to try to get the most reward.
- But by being greedy with respect to estimated values, may not actually get the most reward.

• **Exploitation** - exploit knowledge for short-term benefit



$$\begin{aligned} q(a) &= 3 \\ N(a) &= 5 \\ q_*(a) &= 3 \end{aligned}$$



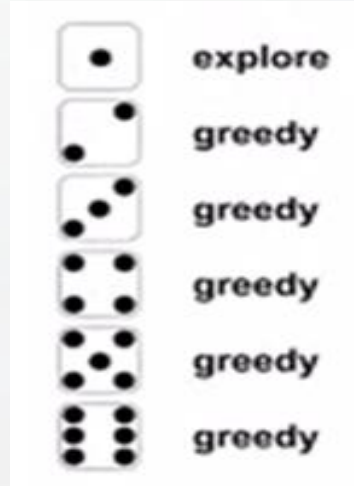
$$\begin{aligned} q(a) &= 0 \\ N(a) &= 0 \\ q_*(a) &= 4 \end{aligned}$$



$$\begin{aligned} q(a) &= 0 \\ N(a) &= 0 \\ q_*(a) &= 2 \end{aligned}$$

Epsilon-Greedy Action Selection

- The ϵ -greedy selection process is given as follows



In this case $\epsilon = 1/6$

$$A_t \leftarrow \begin{cases} \operatorname{argmax}_a Q_t(a), & \text{with probability } 1 - \epsilon \\ a \sim \operatorname{Uniform}(\{a_1 \dots a_k\}), & \text{with probability } \epsilon \end{cases}$$

Counter Example Exploration and Exploitation

- A visiting professor will stay at a city 300 days
- A city has 3 restaurants
- Restaurant 1 has average reward 10 with standard deviation 5
 - Min 5 Average 10 Max 15
- Restaurant 2 has average reward 8 with standard deviation 4
 - Min 4 Average 8 Max 12
- Restaurant 3 has average reward 5 with standard deviation 2.5
 - Min 2.5 Average 5 Max 7.5

Counter Example

Exploration and Exploitation

- Exploration

- Try Restaurant 1 100 days, Try Restaurant 2 100 days, Try Restaurant 3 100 days,
- Total rewards: $100 \times 10 + 100 \times 8 + 100 \times 5 = 2300$

- Exploitation

- Try 3 different restaurants for 3 days
- Get rewards as follows: $R_1 = 7$, $R_2 = 8$, $R_3 = 5$
- Total rewards: $(7 + 8 + 5) + 297 \times 8 = 2396$

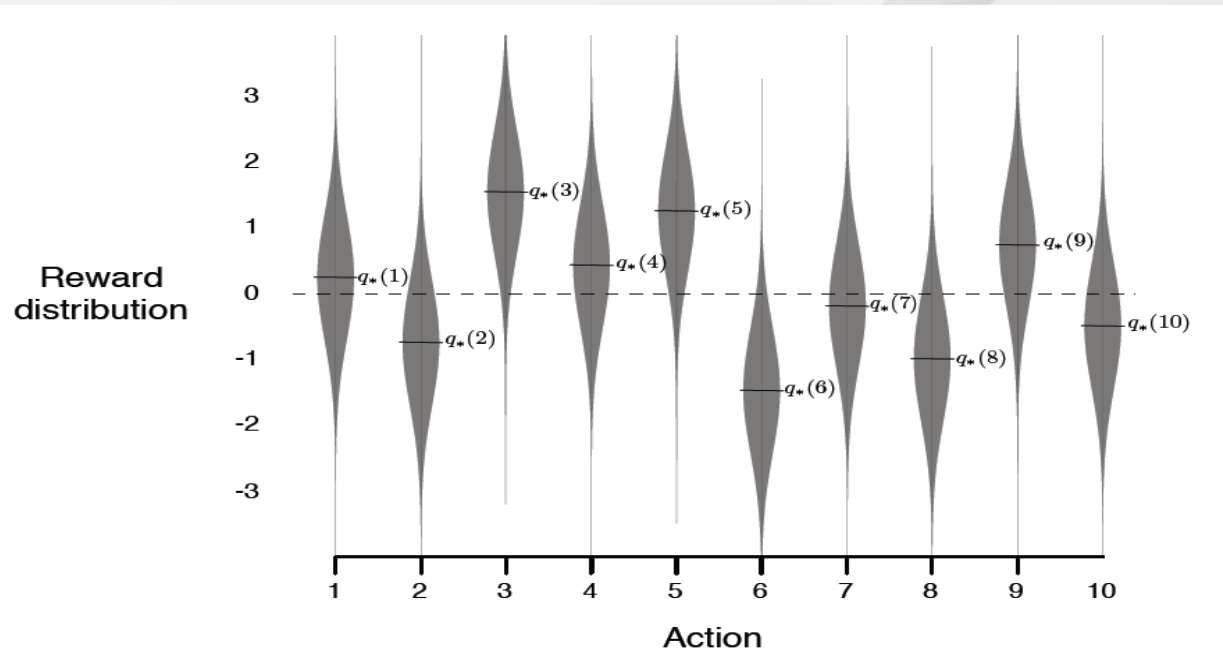
Counter Example

Exploration and Exploitation

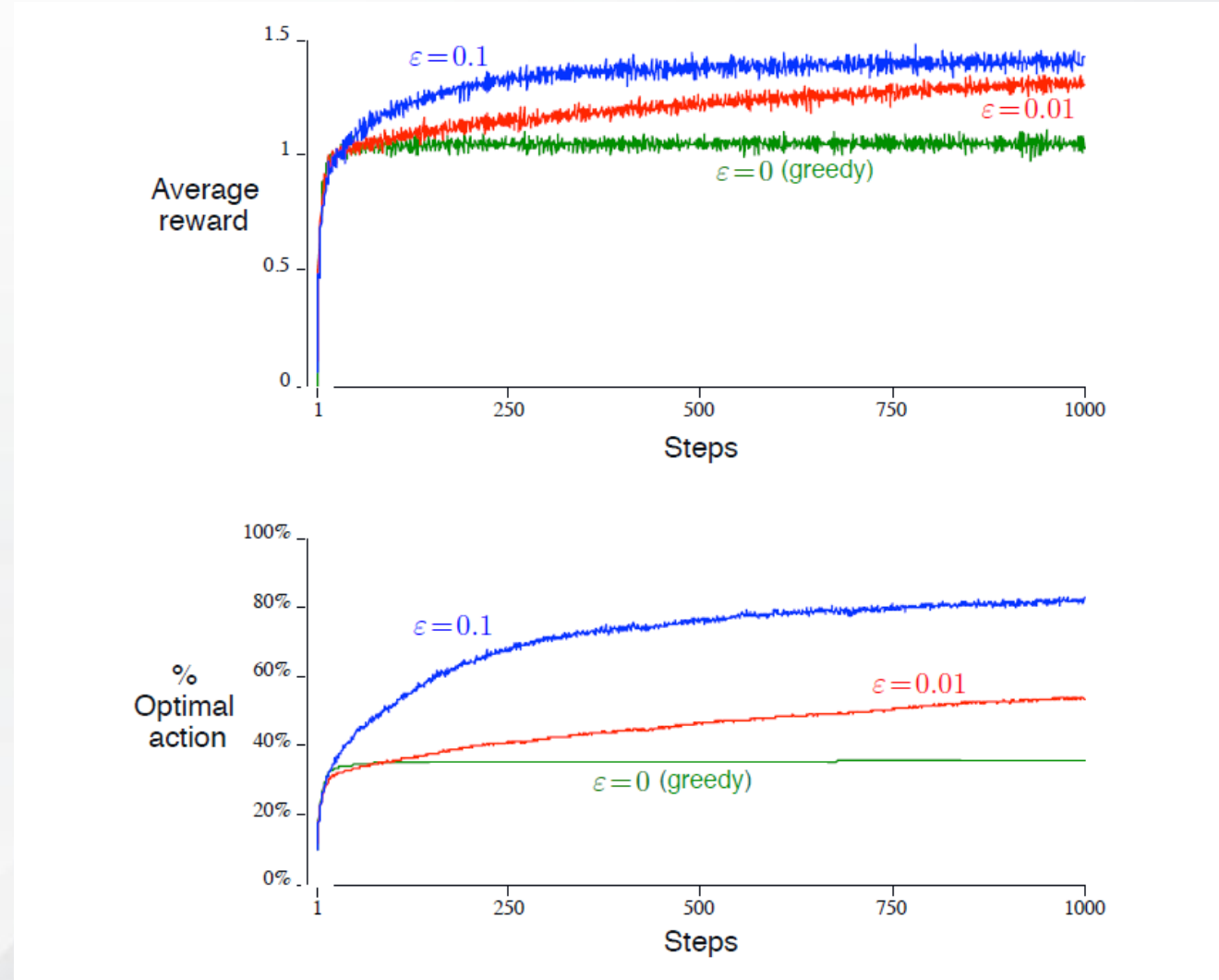
- ϵ - Greedy
 - let $\epsilon = 10\%$ 30 explore and then 270 exploit
 - Total rewards: 2907

The 10-armed Testbed

- In order to understand how to assess the relative effectiveness of the greedy and ε -greedy selection process methods, they were compared numerically on a set of 2000 randomly generated k-armed bandit problems with $k = 10$.
- For each bandit problem, the action values, $q_*(a)$, $a = 1, \dots, 10$, were selected according to a normal (Gaussian) distribution with mean 0 and variance 1. Then, when a learning method applied to that problem selected action A_t at time step t , the actual reward, R_t , was selected from a normal distribution with mean $q_*(a)$ and variance 1.



The 10-armed Testbed



- Read section 2.3 in the book (The 10-armed Testbed)

Optimistic Initial Values (Example: Clinical Trial)

- Previously the initial estimated values were assumed to be 0, which is not necessarily optimistic.
- Let's say that our doctor optimistically assumes that each treatment is highly effective before running the trial.
- To make sure we're definitely overestimating, let's make the initial value for each action 2.
- Let's assume the doctor always chooses the greedy action.

A reward of 1 if the treatment succeeds otherwise 0

$$Q_1(\text{P}) = 2.0$$

$$Q_1(\text{Y}) = 2.0$$

$$Q_1(\text{B}) = 2.0$$

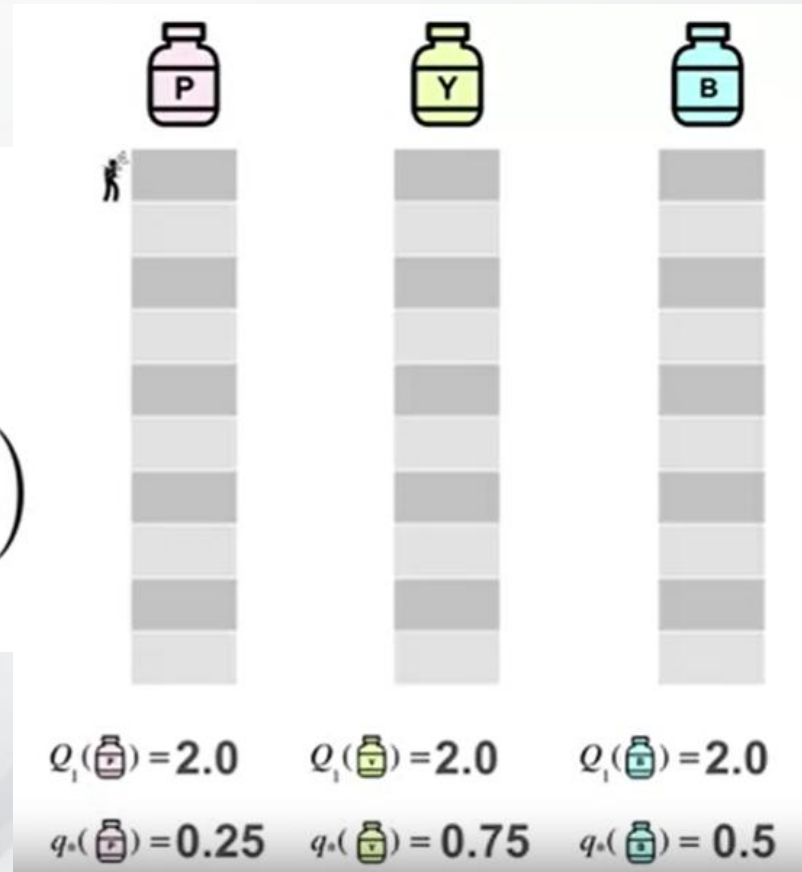


Optimistic Initial Values

A reward of 1 if the treatment succeeds otherwise 0

$$Q_{n+1} \leftarrow Q_n + \alpha(R_n - Q_n)$$

Let $\alpha = 0.5$

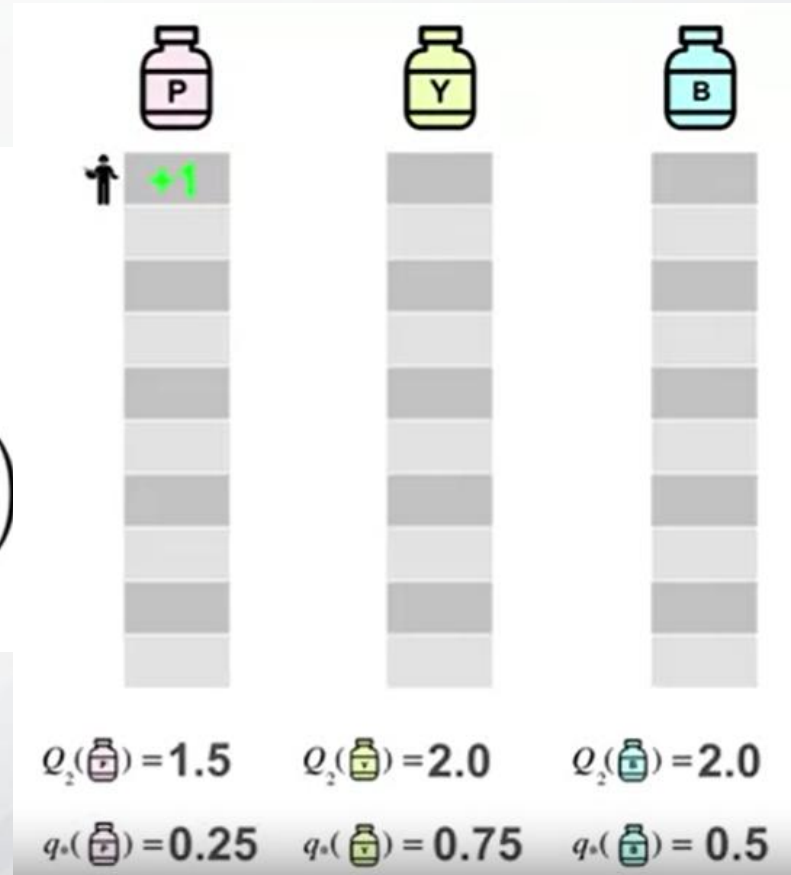


Optimistic Initial Values

A reward of 1 if the treatment succeeds otherwise 0

$$Q_{n+1} \leftarrow Q_n + \alpha (R_n - Q_n)$$

Let $\alpha = 0.5$

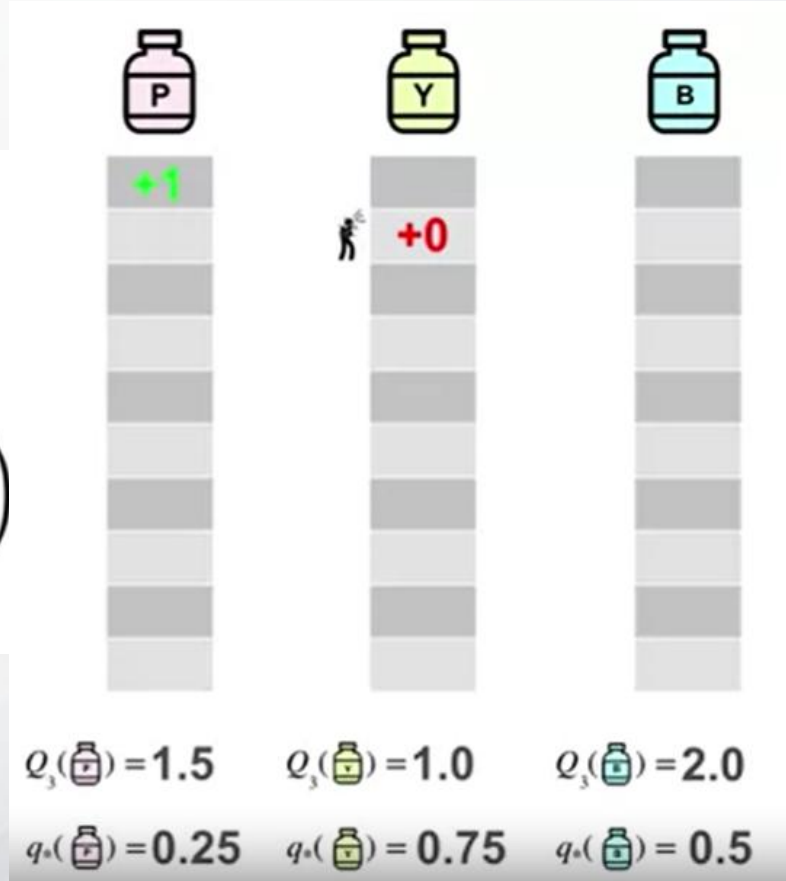


Optimistic Initial Values

A reward of 1 if the
treatment succeeds
otherwise 0

$$Q_{n+1} \leftarrow Q_n + \alpha(R_n - Q_n)$$

Let $\alpha = 0.5$

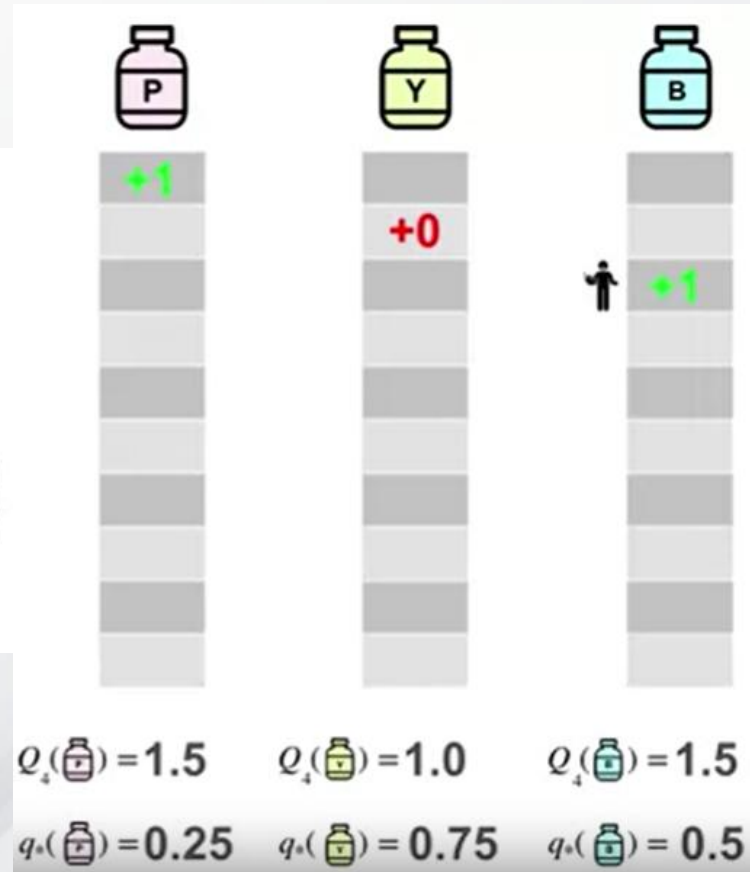


Optimistic Initial Values

A reward of 1 if the treatment succeeds
otherwise 0

$$Q_{n+1} \leftarrow Q_n + \alpha (R_n - Q_n)$$

Let $\alpha = 0.5$

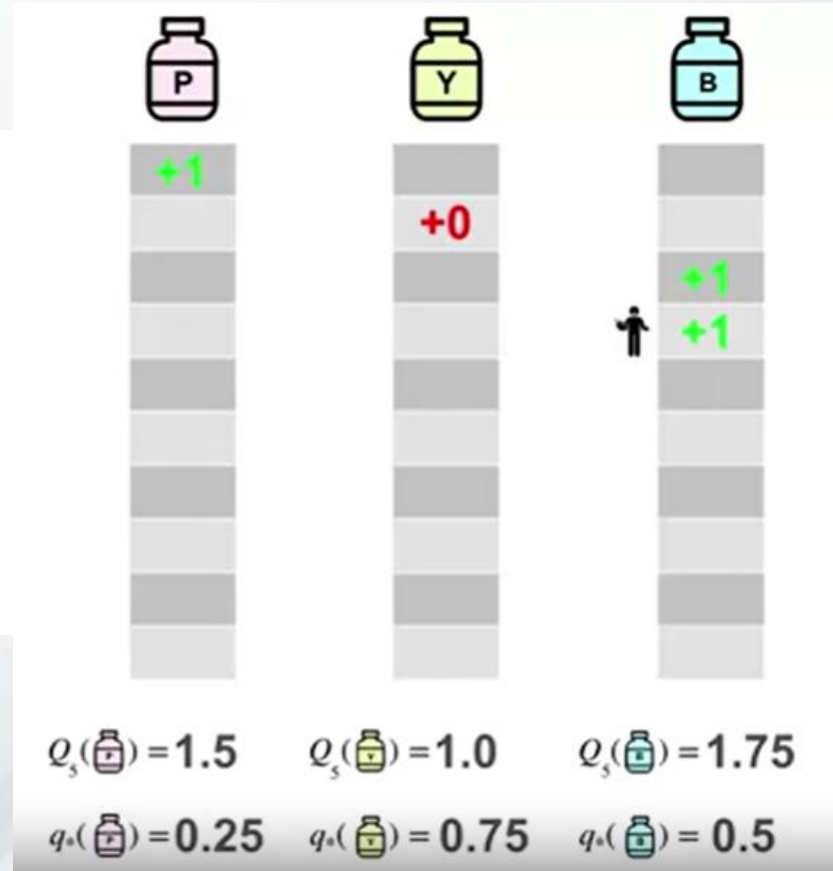


Optimistic Initial Values

A reward of 1 if the treatment succeeds
otherwise 0

$$Q_{n+1} \leftarrow Q_n + \alpha (R_n - Q_n)$$

Let $\alpha = 0.5$

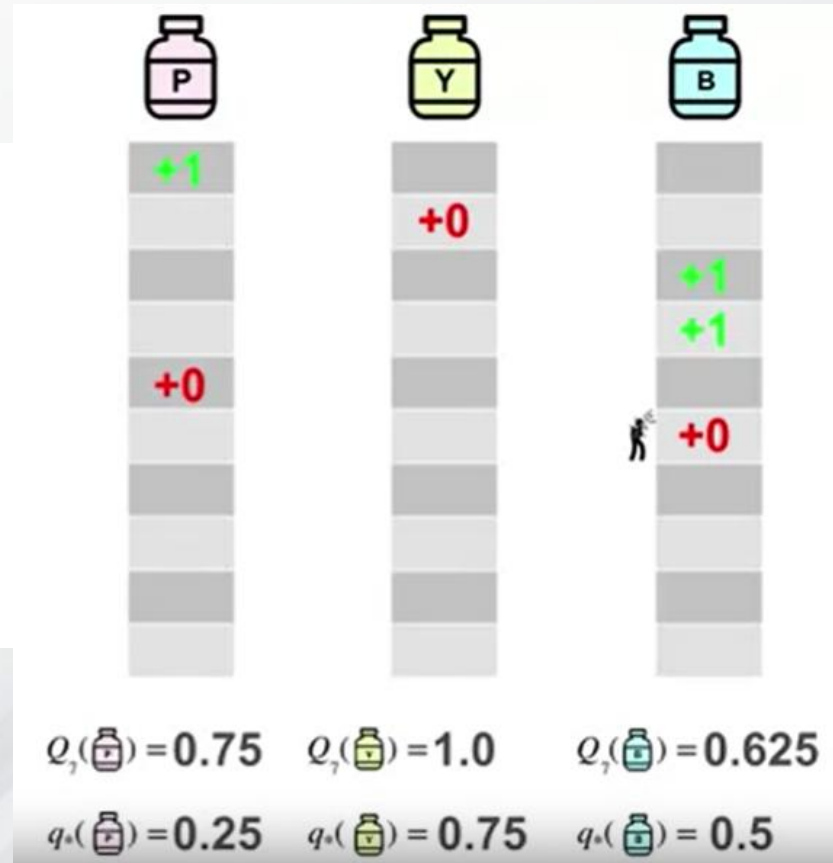


Optimistic Initial Values

A reward of 1 if the treatment succeeds otherwise 0

$$Q_{n+1} \leftarrow Q_n + \alpha(R_n - Q_n)$$

Let $\alpha = 0.5$

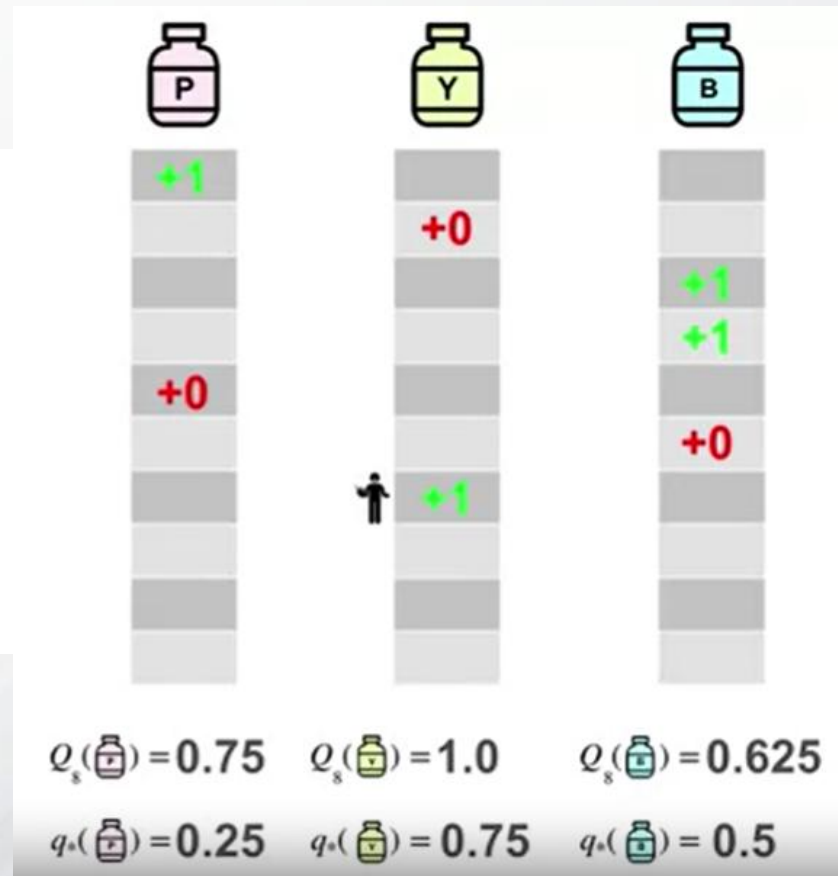


Optimistic Initial Values

A reward of 1 if the treatment succeeds otherwise 0

$$Q_{n+1} \leftarrow Q_n + \alpha(R_n - Q_n)$$

Let $\alpha = 0.5$

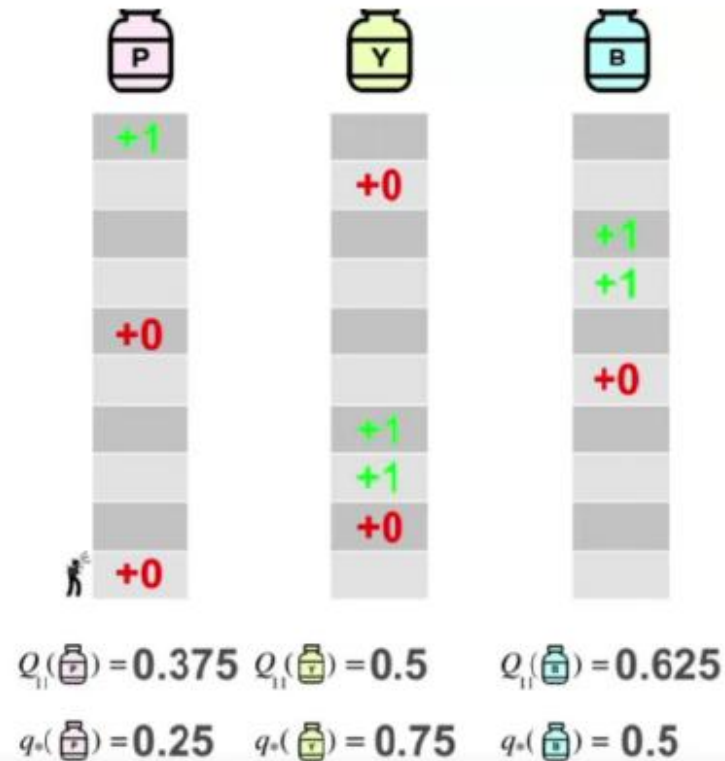


Optimistic Initial Values

A reward of 1 if the treatment succeeds otherwise 0

$$Q_{n+1} \leftarrow Q_n + \alpha(R_n - Q_n)$$

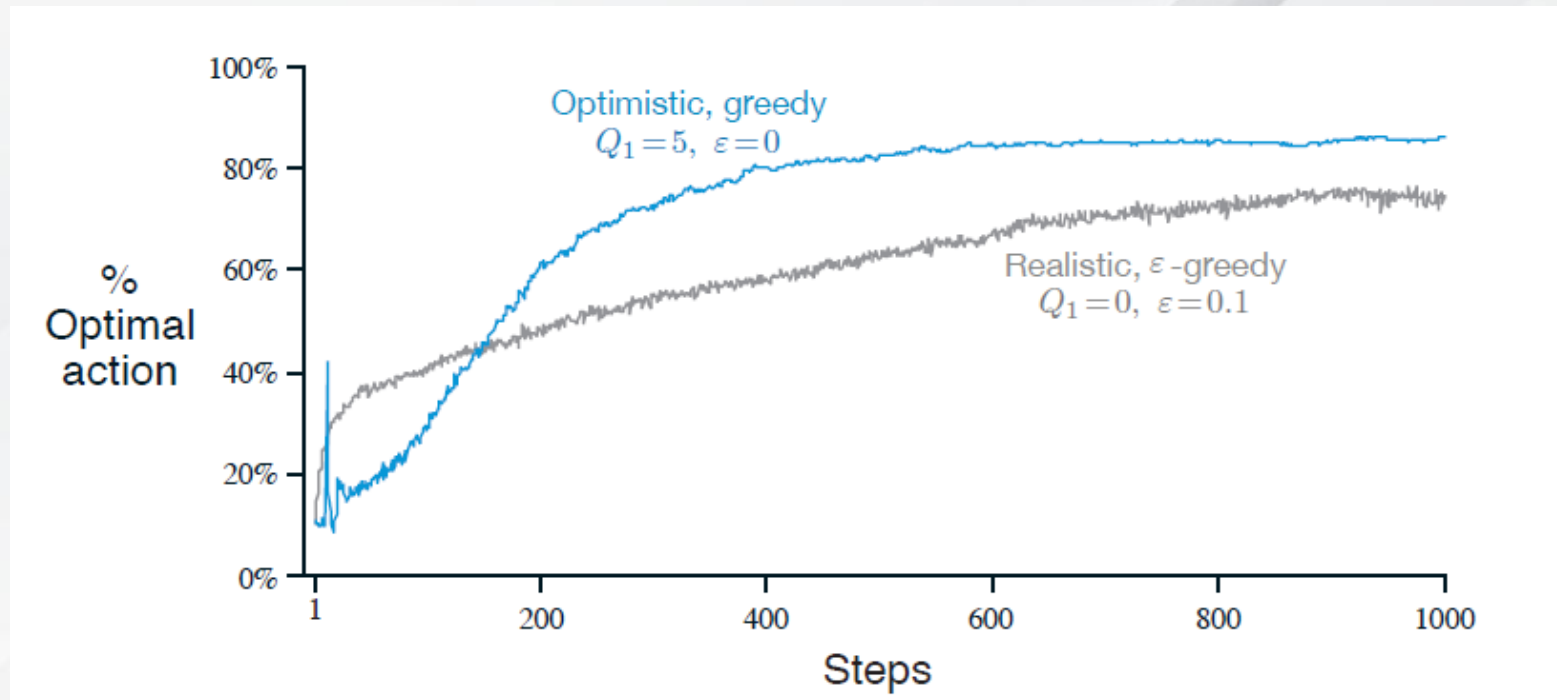
Let $\alpha = 0.5$



- We can see how this optimistic values incentive exploration during the initial steps of the experiment, making Q closer to q^*

Optimistic Initial Values

- The effect of *optimistic initial action-value* estimates on the 10-armed testbed.
- Both methods used a constant step-size parameter, $\alpha = 0.1$.



Limitations of Optimistic Initial Values

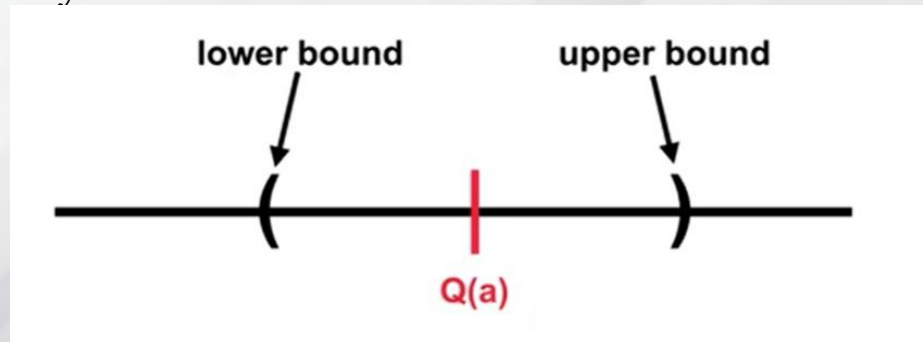
- Some of the limitations of optimistic initial values are
 - ✓ Optimistic initial values **only drive early exploration**
 - ✓ They are not well-suited for **non-stationary problems**
 - ✓ We **may not know** what the optimistic initial value should be

Upper-Confidence Bound (UCB) Action Selection

- Remember the Epsilon-Greedy selection method

$$A_t \leftarrow \begin{cases} \operatorname{argmax}_a Q_t(a) & \text{with probability } 1 - \epsilon \\ a \sim \text{Uniform}(\{a_1 \dots a_k\}) & \text{with probability } \epsilon \end{cases}$$

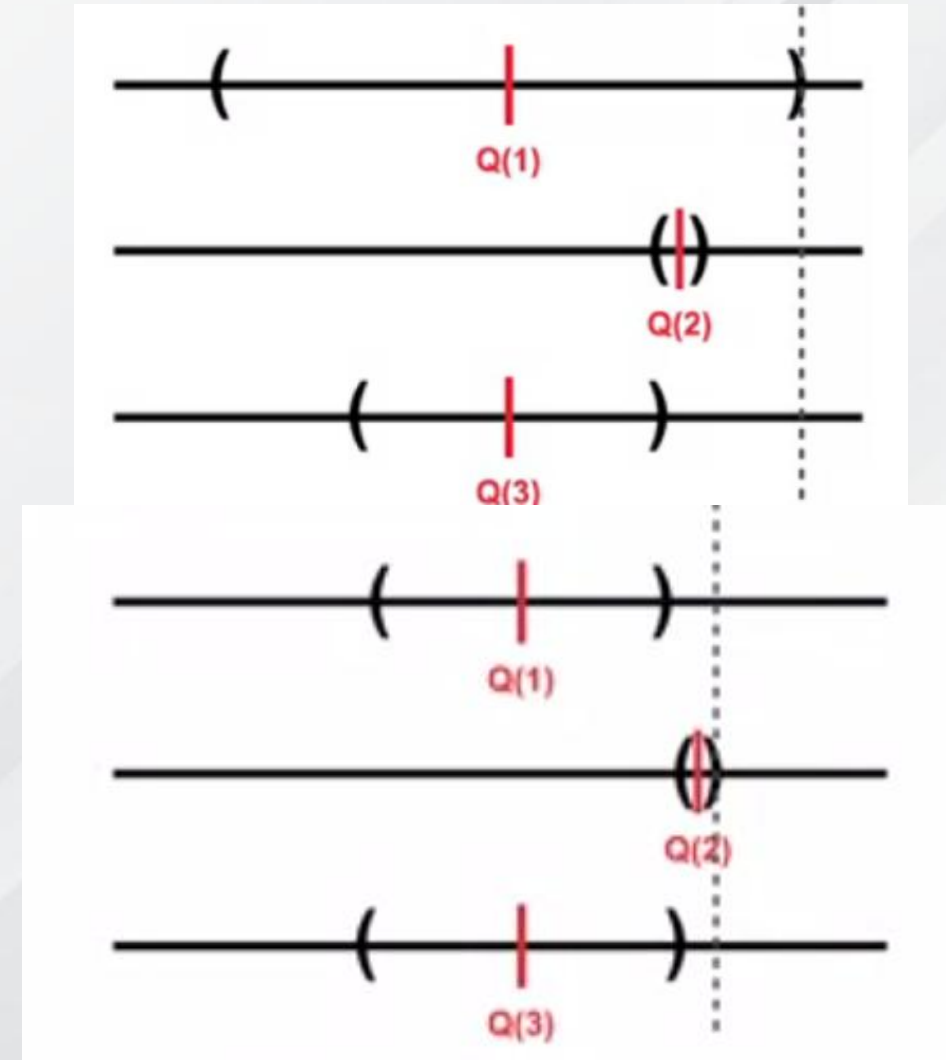
- If we had a notion of uncertainty in our value estimates, we could potentially select actions in a more intelligent way.



These brackets represent a **confidence interval (uncertainty)** around $q_*(a)$.

Optimism in the Face of Uncertainty

- Optimistically picks the action that has the highest upper bound.
- This makes sense because either it does have the highest value and we get good reward, or by taking it we get to learn about an action we know least about.



Upper-Confidence Bound (UCB) Action Selection

- The following equation defines the action selection process based on the Upper-Confidence Bound (UCB).

$$A_t \doteq \operatorname{argmax} \left[Q_t(a) + c \sqrt{\frac{\ln t}{N_t(a)}} \right]$$

- The c parameter is a user-specified parameter that controls the amount of exploration.

$$A_t \doteq \operatorname{argmax} \left[\underset{\text{Exploit}}{Q_t(a)} + c \sqrt{\frac{\ln t}{N_t(a)}} \right]$$

Exploit Explore

Upper-Confidence Bound (UCB) Action Selection

- To see the effects of this action selection criteria in

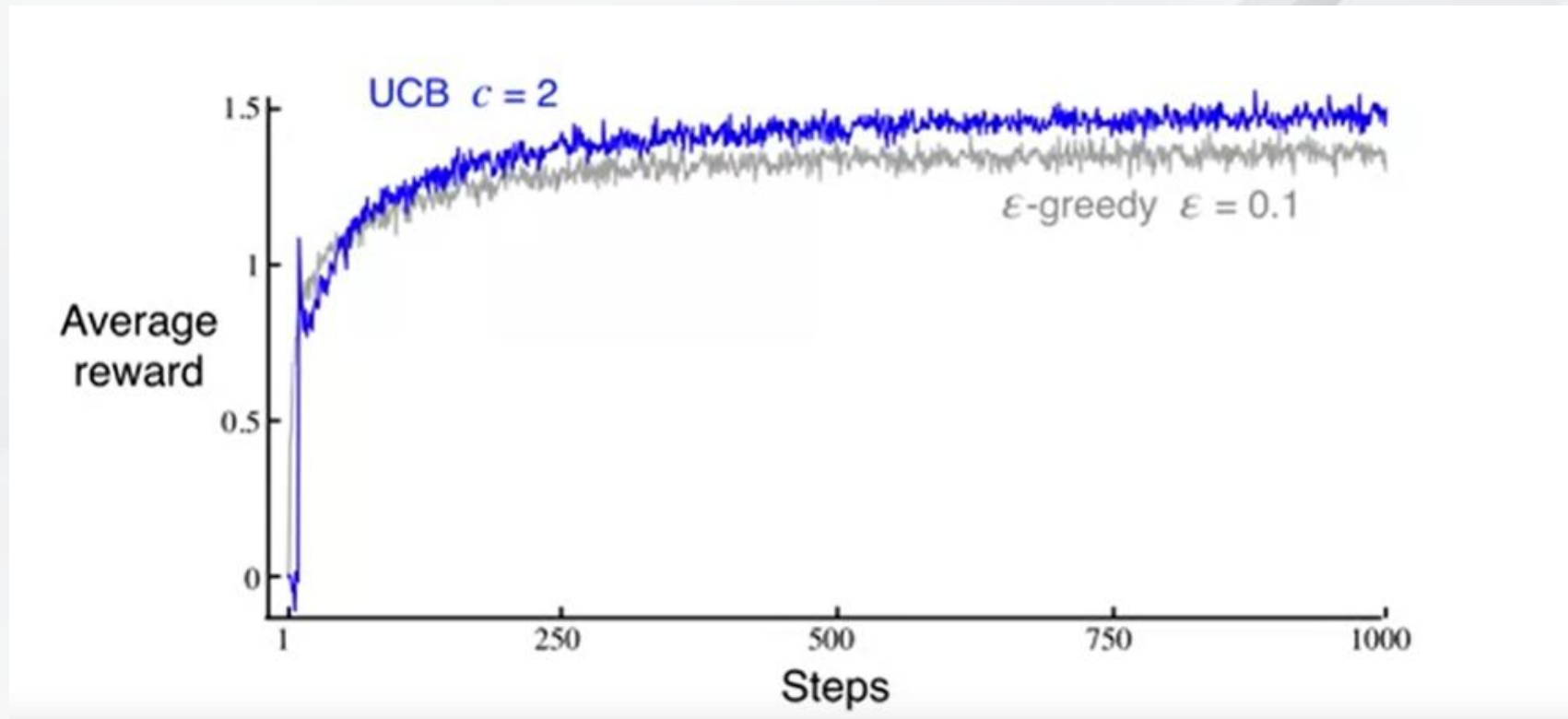
$$A_t \doteq \operatorname{argmax} \left[Q_t(a) + c \sqrt{\frac{\ln t}{N_t(a)}} \right]$$

$$c \sqrt{\frac{\ln t}{N_t(a)}} \rightarrow c \sqrt{\frac{\ln \text{ timesteps}}{\text{times action } a \text{ taken}}} \begin{cases} c \sqrt{\frac{\ln 10000}{5000}} \rightarrow 0.043c \\ c \sqrt{\frac{\ln 10000}{100}} \rightarrow 0.303c \end{cases}$$

- The use of the natural logarithm means that the increases get smaller over time
- All actions will eventually be selected, but actions with lower value estimates, or that have already been selected frequently, will be selected with decreasing frequency over time.

Upper-Confidence Bound (UCB) Action Selection

- The effect of the UCB on the 10-arms TestBed





Thank You